Short Correcion to Theorem of Cook

Juan Manuel Dato Ruiz

jumadaru@gmail.com

Abstract

In the book "Computers and Intractability" of Michael R.Garey & David S. Johnson we have a transcription of the famous Theorem of Cook, showed in 1971. When we see the results of actual studies we will find a contradiction: in a side we could assert problems in NP are not at all in P and, in other hand, we could assert too that there is an NP-complete (SAT) that is in P. We cannot deny we have to solve the contradiction but, where is the error? Scaning this documentation, we will guess my intention is to open a new paradigm in the Maths World, because we accepted in the past a false demonstration.

Explanations

If we want to understand why can I ensure that Cook's Theorem is false, firstly we will have to read his own transcription: The Cook's Theorem, in the refereed book, begins defining the class NP-Complete (page 37) like the set of the NP problems which can be transformed in the rest of NP with a code which does not exceed in resources of time more than polynomially (respecting size of entry). In this way, if a NP-Complete could be solved in "poly-time", then every NP could get an implementation (configuration in a Turing Maching) which costs are "poly-time" too; for getting the knowledge NP=P.

Saying that, the definition sounded very much interesting; but, of course, the objective of Cook was to demonstrate that almost a problem configurable in NP exists with that property. The demonstration of existence of that initial problem NP-complete is the proper Cook's Theorem itself (page 39).

The reasoning done by Mr Cook in this Theorem was to assert the problem of logic satisfiability could be coded in a reasonable scheme with a languaje L_{SAT}. That languaje distingishes the entries which gets a boolean acceptation into the formula it represents. That is, through a Non Deterministic Turing Machine we could solve in a "poly-time" whether the formula described in the languaje satisfies or not.

After presented L_{SAT}, now we will have to ensure that every languaje L in NP could be solved if L_{SAT} was solved before in a concrete bounded time. To get that objetive, we will use a function $f_L(x)$ whose property is to recognize every x in L if and only if f_L contains an assigment that satisfies the boolean formula. We will understand, therefore, that x is an accepted entry by every NP and it is connected to SAT using f. Solving that connection we will solve in that bounded time every NP.

The work of this function will be to map each poosibility in his own Turing Machine and, considering the "poly-time", our objetive will be to demonstrate only the existence of a formula well formed with the same force of the presented problem. And, in fact, the process of demonstration continued in this way: Considering the machine if not deterministic, the

bound will be fixed by a poly got by the size of the entry (page 40), so as Mr. Cook continued as: “This will enable us to describe such a computation completely using only a limited number of Boolean variables and a truth asigment to them”. At this very point everyone could say “yes, in Theory”, Can we ensure every problem NP imagined by us (independientelly of its hardness) will be able to translate to a boolean formula representing the same problem?

When Cook continued with that assertion he mentioned a “poly – quantity” of logic variables completely undefined; from a point of view of mathematician formalism, those variables could be consideared well defined, however we have to get more rigorousity: Does exist a correspondence between the original problem and the formula well formed for constructing? As saying as, is it constructible? More especifically, constructing the formula with every variable then we only will recognize the existence of those variables as formulations in order 0 (without any kind of unification) executing the problem L into the solving machine. That is, if we want to construct the formula well formed before we will have to know the solution of the problem.

This is, basically, the error in the Theorem of Cook: to get constructed the first NP-complete we have to solve before each possible NP problems. Obviously that is unavaliable, because solving the boolean satisfiability we will have no posibilities of finding a code enabled to convert the problem to every other NP.

In other hand, besides the theorem of Cook ensured the existence of that code, I do not know any evidence of itself. That enlighted code will approach every small advances on the boolean validity for solving every NP problem in general; we can imagine what would be happen if that code would exist, speaking about a lot of benefits. So, if we do not understand exactly why the Theorem is false, we will can smell that there is no other way.

Moreover, could not think we in a resolution of a problem of the Principia Mathematica as a bounded problem in time? The Cook’s Theorem does not specify exactly the border. It only referees bound is polynomial (without any kind of expression delimitating the use). Well, let’s change the word polynomial to exponential, or only bounded (it halts). If we have a specification desbribed from axioms of the Principia Mathematica, that specification could be demonstrable in a bounded time by any Turing Machine, to get an omega coherency (almost if we use a reasonable algorithm of unification). Then, if we combine the reasons of the Theorem of Cook with an application on Principia Mathematica, we will always find a formula for satisfying boolean logic of order 0 to satisfy a formulation of the Principia. So, that is a contradiction with the two most known results of Gödel: theorems of completeness and completeless.

That is the reason why the Theorem of Cook cannot be applied in a theoretical point of view.